PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference	**FOR FURTHER ACTION** See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)	
International application No. PCT/GB 03/02784	International filing date *(day/month/year)* 30.06.2003	Priority date *(day/month/year)* 28.06.2002
International Patent Classification (IPC) or both national classification and IPC G06F15/80		
Applicant CRITICAL BLUE LTD et al.		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 9 sheets, including this cover sheet.

   ☒ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

   These annexes consist of a total of 2 sheets.

3. This report contains indications relating to the following items:

   - I ☒ Basis of the opinion
   - II ☐ Priority
   - III ☐ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
   - IV ☐ Lack of unity of invention
   - V ☒ Reasoned statement under Rule 66.2(a)(ii) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
   - VI ☐ Certain documents cited
   - VII ☐ Certain defects in the international application
   - VIII ☐ Certain observations on the international application

Date of submission of the demand	Date of completion of this report
12.01.2004	07.06.2005
Name and mailing address of the international preliminary examining authority: European Patent Office - P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk - Pays Bas Tel. +31 70 340 - 2040 Tx: 31 651 epo nl Fax: +31 70 340 - 3016	Authorized Officer Knapczyk, F Telephone No. +31 70 340-8989

BEST AVAILABLE COPY

I. Basis of the report

1. With regard to the **elements** of the international application *(Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17))*:

Description, Pages

1, 2, 4-29	as originally filed
3	received on 14.02.2005 with letter of 10.02.2005

Claims, Numbers

9-34	as originally filed
1-8	received on 14.02.2005 with letter of 10.02.2005

Drawings, Sheets

1/12-12/12	as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

   These elements were available or furnished to this Authority in the following language: , which is:

   - ☐ the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
   - ☐ the language of publication of the international application (under Rule 48.3(b)).
   - ☐ the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

   - ☐ contained in the international application in written form.
   - ☐ filed together with the international application in computer readable form.
   - ☐ furnished subsequently to this Authority in written form.
   - ☐ furnished subsequently to this Authority in computer readable form.
   - ☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
   - ☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

   - ☐ the description, pages:
   - ☐ the claims, Nos.:
   - ☐ the drawings, sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)).

   (Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Yes:	Claims	1-34
	No:	Claims	
Inventive step (IS)	Yes:	Claims	
	No:	Claims	1-34
Industrial applicability (IA)	Yes:	Claims	1-34
	No:	Claims	

2. Citations and explanations

   see separate sheet

Re Item V

Reasoned statement with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Reference is made to the following documents:
   D1: US-B-6 408 428 (SNIDER GREG ET AL) 18 June 2002 (2002-06-18)
   D2: ATHANAS P M ET AL: "PROCESSOR RECONFIGURATION THROUGH INSTRUCTION-SET METAMORPHOSIS" COMPUTER, IEEE COMPUTER SOCIETY, LONG BEACH., CA, US, US, vol. 26, no. 3, 1 March 1993 (1993-03-01), pages 11-18, XP002067143 ISSN: 0018-9162
   D3: A MARSHALL, T STANSFIELD, I KOSTARNOV, J VUILLEMIN, B HUTCHINGS: "A reconfigurable arithmetic array for multimedia applications" PROCEEDINGS OF THE 1999 ACM SIGDA SEVENTH INTERNATIONAL SYMPOSIUM ON FIELD PROGRAMMABLE GATE ARRAYS, [Online] 21 February 1999 (1999-02-21), - 23 February 1999 (1999-02-23) pages 135-143, XP002305692 MONTEREY, CALIFORNIA, UNITED STATES Retrieved from the Internet: URL:http://delivery.acm.org/10.1145/300000 /296444/p135-marshall.pdf?key1=296444&key2 =6645350011&coll=portal&dl=ACM&CFID=218182 8&CFTOKEN=68827537> [retrieved on 2004-11-12]
   D4: EP-A-0 508 075 (LSI LOGIC CORP) 14 October 1992 (1992-10-14)

2. The amendments brought by the applicant in the subject-matter of claim 1 and in the description on page 3 are satisfying the conditions stated in Article 28(2) PCT. The new (c) statement in the subject-matter of claim 1 is supported by the description on page 7, third paragraph but also on page 8, fourth paragraph and on page 9, second paragraph where it is direct and unambiguous that the execution units are the core of functional units, which can be connected directly together without using a central register.

3. However, the present application does not meet the criteria of Article 33(1) PCT, because the subject-matter of claims 1 and 34 is not inventive in the sense of Article 33(3) PCT.

   3.1. The document D1 is regarded as being the closest prior art to the subject-matter of claim 1 and discloses (the references in parentheses applying to this

document):
A method of automatic configuration of a microprocessor architecture (column 3, lines 18 to 24) whereby:
(a) the architecture includes a configurable number of execution units;
(b) the architecture has configurable connectivity between those execution units; (column 4, lines 39 to 42)
(c) the execution units are able to communicate data directly without the need to be connected between register files that are shared between multiple execution units (figure 2 and column 8, line 51 to column 9, line 12 where it is clear from this example that the functional unit F(+-,1) is directly connected to functional unit F(+-,2) via the interconnects and bus B1 links the output port of the first functional unit to the input port of the second one, the register being considered as a special case of functional units); and
(d) a particular input program is used to influence decisions regarding execution unit replication and connectivity (column 3, lines 33 to 52 where the application program is the input program, column 4, lines 28 to 52 where it is directly and unambiguously derivable that data and control flows of the program are used as a basis to obtain the hardware specification of the candidate architectures and influence their modification, and column 11, lines 59 to 62).

The subject-matter of claim 1 differs from D1 in that the program structure, namely the data and control flows, is used to influence decisions regarding execution unit replication and connectivity, whereas D1 uses statistical information to influence the architecture after having mapped the code to the architecture. The technical effect provided is that the design of the first candidate processor is made out of the program structure, and therefore better suited to the program than the first architecture used in D1.

The problem to be solve may therefore be regarded as how to obtain efficiently a better processor architecture.

The skilled person knows that using a dataflow graph derived from a program source code allows to realise the synthesis of a first hardware architecture which is more efficient than a non-specific first architecture (see document D2, page 15, left-hand column, line 8 to middle-column, line 12). The use of the data and

control flow of an input program to elaborate a hardware architecture therefore solves the problem at hand. It is in the capacity of the skilled person to incorporate the teaching of D2 into the system of D1 thus arriving at the subject-matter according to claim 1 without the exercise of inventive skills.

Therefore claim 1 is not inventive.

3.2. For the reasons presented above, claim 34 is not inventive, the step from the method of claim 1 to the apparatus of claim 34 being trivial.

4. Moreover, the dependent claims 2 to 33 do not contain additional features, which meet the requirements of the PCT in respect of inventive step (Article 33(3) PCT) for the following reasons:

4.1. For claims 2, 3, 5 and 6, the document D1 also describes that multiple candidate architectures are generated (column 3, lines 20 to 24), best architectures are automatically selected to satisfy design constraints where it is implicit that these constraints are user defined (column 3, lines 22 to 24 and column 3, line 65 to column 4, line 3), new connections and functional units are added to the architecture on each generation (column 11, lines 59 to 67 and column 3, lines 49 to 52), the code is mapped in the trial architecture to influence connectivity (column 3, lines 33 to 52 and column 11, lines 59 to 67 where it is directly and unambiguously derivable that if a new architecture with added components is proposed, connectivity will be modified).

4.2. Claims 4 and 7 are not inventive because the output of statistics on candidate architecture efficiency is already present in document D1, column 14, lines 1 to 7 and displaying information on a graph is a common design choice. The feature of an execution units conflict influencing the addition of functional units is present in D1 in column 12, lines 12 to 16 where the exclusion added is the conflict which is overcome by adding functional units for each exclusive operation group on column 30, lines 58 to 67; the resulting delays are a common consequence in such architecture conflicts.

4.3. Regarding claims 8 and 9, document D1 also discloses that every candidate

generated contains a minimum set of execution types and a minimum connectivity between all execution units (figure 2 and column 8, lines 51 to 63).

4.4. For claims 10 to 13, the possibility to run less efficiently new code sequences in a processor architecture optimised for a given code is a consequence of the realisation of the optimisation using a minimum set of execution unit types and a minimum connectivity to bind them. Moreover, reachability analysis is a common technique in the art (see for instance document D4, page 4, lines 27 to 34). Therefore claims 9 to 13 do not involve an inventive step.

4.5. Claim 14 is not inventive because the determination of the initial connectivity within a processor architecture determined from data flows within graph representations of the input program is a common knowledge the skilled person would use without the exercise of inventive activity (see for instance document D2 page 15, left-hand column, line 16 to middle column, line 17).

4.6. Referring to claims 15 and 16, document D1 also discloses that new connections are added as requested during code generation/compilation (column 3, lines 45 to 51 where it is implicit that a new candidate will have a new set of functional units and connections according to column 3, line 66 to column 4, line 42), the addition of connection is constrained by rules (column 27, lines 60 to 63),

4.7. Claims 17 and 18 are not inventive because the ports number tends to be minimised in every optimised architecture (as it is also the case in document D1, column 33, lines 13 to 19 and column 36, line 65 to column 37, line 20) and so do the changes to candidate architecture to improve performances while staying in the frame of the constraints (column 14, lines 10 to 12 and column 11, lines 59 to 62).

4.8. Claims 19 to 21, 24 and 25 are not inventive because the positioning of a functional unit in a logical grid layout is known from the person skilled in the art as a common practise (see for example D2, page 15, middle column, lines 9 to 19 where an FPGA is a logical grid layout). In the field of processor design, distance means latency so it is obvious that a maximum distance between the units constraints the positioning of the elements. The obtention of execution units from

component libraries including a reference to the hardware description of the execution unit is also known in the art (see document D2, page 15, left-hand column, lines 51 to 54). Putting the register in a position where the connection distances will be minimalised is one of the straightforward possibilities the skilled would use to reduce the latency and the connection needed (see for example document D3, page 3, right-hand column, lines 7 to 21 where the RAM blocks containing data buffers and linked to the grid of ALUs are spread over the ALU grid), therefore putting a register in the centre of the grid is not inventive. For the same reason, limiting latency by placing closer from the register the elements which are more often used does not involve an inventive step (especially as document D1 describes a statistical usage on how often the component is used (column 11, lines 59 to 60) in order to find the best performance architecture (column 14, lines 10 to 12)).

4.9. Claim 22 is not inventive because document D1 also discloses that component have precharacterised characteristics (column 36, line 63 to column 37, line 20 where mini-MDES store the units characteristics and column 5, lines 2 to 7).

4.10. Claim 23 is not inventive because it is the basic concept of processor design to have an optimised operational frequency for a given implementation. See for instance document D1, which is using the execution speed and the circuit complexity as design objectives (column 4, lines 1 to 3). These are parameters which directly influence the resulting operational frequency.

4.11. Claim 26 is not inventive because the architecture optimisation for certain functions specified to the system is known in the art (see document D2, figure 1 where function B is specified as too complex to be optimised in the hardware architecture).

4.12. For claims 27 to 31, document D1 also describes the description file in a hardware description language indicating the hardware construction (column 3, lines 52 to 57 and column 23, lines 60 to 66), the execution word layout to control the execution units (column 22, lines 5 to 20 where the instructions from the different macrocells compose the execution word), the detail of the selection codes associated with operand inputs, outputs registers and execution unit

operation code (column 19, lines 40 to 52 and figure 1).

4.13. Claims 32 and 33 are not inventive because the use of the weighting of functions to influence the weighting of individual instruction is known in the art (see for instance document D1, column 12, lines 25 to 29 where the cost of an expensive function is taken into account into the design in order to optimise its use).

SUMMARY OF INVENTION

If an application specific processor is being synthesized then far greater efficiency can be obtained by determining the functional unit and connectivity resources on the basis of the particular application it is to be used for. The mix must be determined automatically by analyzing the type of code that the processor is going to execute. It is not sufficient to just measure the frequency of usage of different operations and replicate units on that basis. There is no advantage to be gained unless there are situations in which a number of replicated functional units could be usefully used in parallel. The optimisation must perform detailed analysis of the data flows within the program (especially the most performance critical parts) and use that to determine the right number of, and connectivity for, the functional units.

An initial candidate architecture is created with a minimal connectivity network and one copy of each functional unit. The functional units can communicate data directly to one another. Software code is then targeted at the candidate architecture. As such code is being generated, information is collected about resources that could be used to improve the code generation. For instance, a count is kept of the number of occasions in which data needs to be transported between two functional units, and there is not a direct connection between them. A count is also maintained of the number of times that all functional units of a particular type are busy with existing operations when another operation of that type needs to be performed.

The counts produced during code generation are then weighted by two different factors. Firstly, code from the software functions that have been marked as being critical for overall performance are given a higher weighting. Instructions within the inner loops of such functions are given a still higher weighting. The weightings are used direct the allocation of new resources. .

A new resource might be a duplicate functional unit or a new connection between functional units. The area overhead of the new resource is compared against its weight in comparison to other potential resource additions. A choice is made for a new resource taking into account both the area it occupies and its potential benefit. When a new resource has been added to the architecture the code generation process is repeated. The addition of the resource should improve the performance of the architecture and also reveal what further resources should be added.

CLAIMS

1. A method of automatic configuration of a microprocessor architecture whereby:

   (a) the architecture includes a configurable number of execution units;

   (b) the architecture has configurable connectivity between those execution units;

   (c) the execution units are able to communicate data directly without the need to be connected between register files that are shared between multiple execution units; and

   (c) the data and control flows within a particular input program are used to influence decisions regarding execution unit replication and connectivity..

2. The method according to claim 1 whereby multiple candidate architectures are generated.

3. The method according to claim 2 whereby the best candidate architecture is automatically selected on the basis of user defined metrics.

4. The method according to claim 2 whereby data is output to allow the construction of a graph representing the characteristics of certain candidates.

5. The method according to claim 2 whereby a number of new connections and or execution units are added to the architecture on each generation.

6. The method according to claim 5 whereby mapping of code onto a trial architecture is used to influence connectivity choices.

7. The method according to claim 6 whereby the delays caused by execution unit conflicts in the schedule are used to increase the chances of an additional execution unit of that type being added to the architecture.

8. The method according to claim 1 whereby every candidate generated contains a certain minimum set of execution unit types.